

Optimización de Analizadores para Gramáticas de Unificación de amplia cobertura utilizando la Técnica de Restricción

Arantxa Díaz*
jipdisaa@si.ehu.es

Nora La Serna*
jiblapan@si.ehu.es

Horacio Rodríguez+
horacio@lsi.upc.es

*Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática
Universidad del País Vasco/Euskal Herriko Unibertsitatea

+ Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática
Universidad Politécnica de Catalunya

1. Introducción

En este artículo se describe la metodología seguida en el estudio realizado para asegurar la eficiencia del proceso de análisis de frases con gramáticas de unificación de amplia cobertura. La técnica utilizada para mejorar el proceso de análisis es la técnica de restricción [SHI85], la cual ha sido reconocida como una operación importante para obtener analizadores eficientes con gramáticas de unificación. El estudio consistió en la realización de diversos tipos de experimentos para dar con el objetivo central de la investigación, que como más adelante se explica es seleccionar restrictores adecuados al utilizar la técnica de restricción. El tipo de analizador utilizado para el análisis de las frases es el analizador tabular de Earley para gramáticas de unificación.

Aunque la extensión de las técnicas más conocidas de análisis sintáctico para gramáticas independientes del contexto al caso de las gramáticas de unificación, es en general bastante sencilla, la utilización del material lingüístico asociado a los rasgos de la gramática para guiar el proceso de análisis presenta algunos problemas, que Shieber en [SHI85] señala. El más importante de tales problemas, es que ciertas gramáticas de unificación presentan dominios infinitos de la información gramatical, dando lugar a que, con éstas gramáticas, analizadores estándar podrían ser ineficientes o no terminar el proceso del análisis de una frase. La técnica de restricción pretende soslayar estos problemas.

Adicionalmente, restricción se presenta como un recurso potente para trabajar con Formalismos Gramaticales de Unificación, con muchas aplicaciones en esta área. Puede ser aplicado en analizadores para formalismos como Gramáticas de Cláusulas Definidas (DCG), Gramáticas Léxico Funcionales (LFG) y Gramáticas de Estructura de Frase

Generalizadas (GPSG); puede ser usado en otros algoritmos, como LR; para optimización de analizadores, etc.

En la técnica de restricción, propuesta por Shieber para extender el algoritmo Earley para tratar con gramáticas de unificación, usa el *restringidor*, subconjunto de las estructuras de rasgos pertenecientes a las categorías complejas de la gramática, el que será aplicado en la predicción del algoritmo. Si la gramática usada presenta un número potencialmente infinito de estructuras de rasgos, la técnica aplicada en esta gramática con un restringidor apropiado divide el dominio de la gramática hacia un juego finito de clases equivalentes, esto garantizará que el algoritmo del proceso del análisis termine correctamente¹.

Sin embargo, hay una cuestión abierta al utilizar restricción, no están dados los principios de cómo seleccionar el restringidor para obtener eficiencia en el proceso del análisis, está comprobado que una selección no acertada del restringidor afecta la eficiencia del analizador. En este artículo presentamos el estudio realizado para seleccionar restringidores adecuados en gramáticas de unificación. Además el propósito del estudio es aplicarlo a gramáticas reales y de amplia cobertura.

La organización del artículo es la siguiente: en el punto 2, se presentan las herramientas básicas utilizadas en este estudio; se plantea una metodología para la investigación en el punto 3; en 4, se explican los tipos de experimentos realizados y el análisis de los resultados obtenidos; finalmente, en el punto 5 se presentan las conclusiones.

2. Herramientas básicas utilizadas

2.1 La gramática PATR-II de amplia cobertura

PATR-II es un eficaz, simple y el más común de los Formalismos Gramaticales basados en unificación. Fue desarrollado por S. Shieber [SHI86]. Las componentes tanto léxicas como gramaticales del formalismo se representan mediante estructuras de rasgos, ver en Fig. 1. La estructura de rasgo es en realidad un DAG (directed acyclic graph) que es una función parcial desde rasgos a sus valores, por ejemplo, en Fig. 1, el rasgo número tiene asignado el valor singular; el valor de un rasgo puede ser otra estructura de rasgo como el valor del rasgo concordancia, en Fig. 1. Un DAG puede tener un dominio vacío, llamado DAG variable. Una secuencia de rasgos es llamado camino, por ejemplo <concordancia número>.

¹ Una definición más precisa de esta técnica se presenta en 2.3.

Las reglas gramaticales, ver en Fig. 2, describen la relación entre las cadenas de símbolos terminales y no terminales del lenguaje y las estructuras de rasgos; cada regla PATR-II está formada por una regla libre del contexto y un juego de ecuaciones. Dos operaciones importantes en estos formalismos son la subsunción y la unificación.

1) [categoría : np	2) $x_0 \rightarrow x_1 x_2$
concordancia : [número : singular	$\langle x_0 \text{ cat} \rangle = s$
persona : tercera]]	$\langle x_1 \text{ cat} \rangle = np$
	$\langle x_2 \text{ cat} \rangle = vp$
	$\langle x_0 \text{ head} \rangle = \langle x_2 \text{ head} \rangle$
	$\langle x_0 \text{ head subject} \rangle = \langle x_1 \text{ head} \rangle$

Figura 1: Un ejemplo de estructura de rasgos

Figura 2: Un ejemplo de regla gramatical

La gramática PATR-II de amplia cobertura utilizada en los experimentos ha sido generada a partir de la gramática de unificación objeto del sistema ANLT (Herramientas del Lenguaje Natural ALVEY)². La gramática objeto ALVEY es una variante sintáctica de las gramáticas DCG (Pereira & Warren, 1980), permite definir una amplia cobertura de construcciones gramaticales sintácticas tales como oraciones de tipos declarativos, imperativos e interrogativos, oraciones de relativo y completivas; estructuras coordinadas, comparativos nominales y adjetivales, oraciones pasivas, partitivas, etc.

La gramática PATR-II obtenida para nuestros experimentos resulta de la conversión de 350 de las 450 reglas que contiene la versión cuatro del ANLT. Las reglas convertidas contienen casi todas las reglas de estructuras sintácticas, se han excluido sólo las reglas de conjunción. Hemos convertido 5008 entradas del léxico ANLT al formato PATR-II.

2.2 El Analizador tabular Earley para Gramáticas de Unificación

El analizador de Earley ha sido muy utilizado en el área del Procesamiento del Lenguaje Natural, debido a un excelente comportamiento computacional (presenta un coste temporal cúbico en el peor de los casos). El algoritmo de J. Earley [EAR70] es un algoritmo top-down general y eficiente para analizar gramáticas de contexto libre; el que analiza una cadena de entrada $X_1..X_n$ de izquierda a derecha mirando algún número k de símbolos

² Las herramientas del Lenguaje Natural Alvey, ANLT [GRO93] constituyen un sistema lingüístico computacional desarrollado para producir análisis morfológico, sintáctico, y semántico de una amplia cobertura de frases para el Inglés.

anticipados de la cadena. Para cada símbolo X_i analizado se contruye un conjunto de estados que representan la condición del proceso de reconocimiento en un punto dado del análisis. Habrá tantos conjuntos de estados como elementos tenga la cadena que se está analizando. Para cada paso se realizan tres operaciones: el predictor, el analizador y el completador. El análisis termina cuando se encuentra el estado inicial, si no se llega a este estado la frase no es reconocida por la gramática.

El Analizador tabular del estilo Earley para gramáticas de unificación, en el que Shieber [SHI85] incluye la técnica de restricción, se construye a partir del algoritmo Earley original con dos modificaciones adicionales: primero, en el algoritmo original cuando un nuevo elemento va a ser agregado al chart o conjunto de estados, se verifica si ese elemento está en el chart, en este caso la verificación se realiza por igualdad; mientras que en gramáticas de unificación se realiza por subsunción, es decir si existe un elemento en el chart que subsume al nuevo elemento entonces este último no se incluye en el chart. Segundo, Shieber incluye la técnica de restricción en el paso predictor del algoritmo.

En nuestro estudio hemos utilizado el analizador Unicorn implementado por Gerdemann y Hinrichs en 1988. Es un analizador tabular tipo Earley para gramáticas de unificación, que incorpora la técnica de restricción, y optimizaciones desarrolladas por Gerdemann [GER91] que permiten utilizar la técnica a otras partes del analizador: en la prueba de subsunción de la operación de predicción, y en la operación del completador.

2.3 La Técnica de Restricción³

Sea R un conjunto finito de caminos, llamado *el restrictor*, D un DAG de una gramática de unificación. RD será el DAG restringido de D por el restrictor R si reúne las siguientes condiciones :

1. RD subsume a D , es decir es más general que D ,
2. Para cada camino p en RD , existe un camino q en R , tal que p es un prefijo de q , por lo tanto $RD(p)$ puede no estar definido ó puede tener un valor atómico ó estar específicamente permitido por el restrictor;
3. RD es el DAG más específico que reúne las condiciones anteriores.

³ Para una definición más formal y amplia de la técnica de restricción, ver [SHI85]. La definición presentada en este apartado, es tomado del trabajo de Seiffert [SEI87].

Los restrictores no deben contener un número infinito de caminos diferentes (también llamados clases equivalentes). El siguiente ejemplo ilustra la definición :

Sea el restrictor $R = \{ \langle a \ b \rangle, \langle d \ e \ f \rangle, \langle d \ i \ j \ f \rangle \}$; D , el DAG de la gramática; RD es el DAG restrictivo de D por el restrictor R :

$$D = \left[\begin{array}{l} a : \left[\begin{array}{l} b : c \\ e : (1)[f : [g : h]] \end{array} \right] \\ d : \left[\begin{array}{l} i : [j : (1)] \\ k : 1 \end{array} \right] \end{array} \right] \quad RD = \left[\begin{array}{l} a : \left[\begin{array}{l} b : c \\ e : (1)[f : []] \end{array} \right] \\ d : \left[\begin{array}{l} i : [j : (1)] \end{array} \right] \end{array} \right]$$

3. Metodología empleada para seleccionar restrictores

3.1 Un modelo estadístico para obtener propiedades de instanciación de los rasgos

En la primera etapa del estudio acerca de la selección del restrictor adecuado [LAS95], se estableció la hipótesis: que cuanto más instanciado el rasgo de un restrictor sería buen candidato el rasgo para formar restrictores adecuados; por lo tanto es importante saber qué rasgos asociados a las categorías de la gramática están instanciados durante el análisis de las frases. En la segunda etapa, el estudio ha consistido en comprobar la hipótesis establecida en la primera con la gramática de amplia cobertura PATR-II. Para ello, se definió un modelo estadístico que nos permitiera conocer la situación (probabilidad) de instanciación de los distintos (270) rasgos de la gramática, a lo largo del proceso del análisis.

El modelo se justifica por dos razones: a) No todas las reglas se aplican con igual probabilidad. Por esta razón en el modelo se consideraron la probabilidad de aplicación de cada regla de la gramática; b) Para rasgos de la gramática que no están instanciados, cabe la posibilidad de que los valores de esos rasgos en las reglas, se puedan filtrar de la categoría madre o de categorías que son hermanos.

Los rasgos con probabilidad alta de instanciación, fueron seleccionados inicialmente como restrictores en los experimentos para comprobar si efectivamente son restrictores adecuados. La elaboración del modelo da lugar a un sistema de ecuaciones lineales, que contiene un número igual de variables (probabilidades de cada rasgo y categoría que se pretende hallar) y ecuaciones por resolver. Presentamos formalmente el modelo (Fig. 3):

$$P_{inst}(C_i, R_j) = \sum_k (P(r_k) * (instanciado(C_i, R_j, r_k) + (P(C_{mk}, R_j) + \sum_p P(Ch_k, R_j)) * no_instanciado(C_i, R_j, r_k)))$$

Figura 3: El modelo estadístico para estimar la instanciación de los rasgos.

donde: $P_{inst}(C_i, R_j)$ es la probabilidad del rasgo j , de la categoría i , aparezca instanciado; C_i , categoría i -ésima; R_j , rasgo j -ésima; r_k , regla k -ésima; C_{mk} , categoría madre de la regla k -ésima; Ch_k , categoría del hermano de la regla k -ésima.

La explicación de la fórmula es la siguiente: Hallar la probabilidad de que el rasgo j , de la categoría i , aparezca instanciado, implica buscar en la parte derecha de todas las reglas de la gramática. Para cada regla, en donde está definido el rasgo del que se quiere hallar la probabilidad, se realiza lo siguiente:

1. Si el rasgo j , de la categoría i , está **instanciado** en la regla k -ésima, la matriz representada por *instanciado* (C_i, R_j, r_k) contendrá el valor uno, y cero en el caso contrario. Cuando la matriz contiene el valor uno, entonces $P_{inst}(C_i, R_j, r_k)$ es igual a la probabilidad de la regla k .
2. Si el rasgo j , de la categoría i , **no está instanciado** en la regla k -ésima, la matriz representada por *no_instanciado* (C_i, R_j, r_k) contendrá el valor uno, y cero si está instanciado. En éste paso, cuando la matriz contiene el valor uno, se mira si el rasgo no instanciado puede ser heredado en algún paso del análisis, desde la categoría representada en la parte izquierda de la regla (categoría madre) o a partir de las categorías anteriores a la categoría en estudio en la misma regla (categorías hermanos).

Si el valor del rasgo, que es analizado, se hereda entonces $P_{inst}(C_i, R_j)$ es igual a la probabilidad de la regla, por la probabilidad de la categoría madre o hermano(s), según sea el caso, es decir igual a $P(r_k) * (P(C_{mk}, R_j) + \sum_p P(Ch_k, R_j))$.

Finalmente, si el rasgo j , de la categoría i , en la regla k -ésima es no instanciado, y el valor del rasgo no puede ser heredado, entonces $P_{inst}(C_i, R_j)$ es igual a cero en esa regla.

3.2 Estimación de los parámetros del modelo

De la fórmula se deduce que, para estimar los parámetros del modelo en la gramática de amplia cobertura, son necesarias dos tareas: 1) hallar la probabilidad de aplicación de cada regla de la gramática, y 2) determinar la probabilidad de que un rasgo aparezca instanciado.

Para el cálculo de la **probabilidad de aplicación de las reglas**, dos pasos fueron desarrollados: i) Las reglas aplicadas, las que son usadas para el cálculo, fueron obtenidas desde el proceso del análisis. Para ello, se utilizaron las frases del corpus del ALVEY, y los análisis de las frases se hicieron con el analizador del ALVEY. ii) Se determinaron las probabilidades de aplicación de cada regla de la gramática, cada una de éstas probabilidades resulta de dividir, el número de las aplicaciones de cada regla, y el número total de reglas aplicadas.

Dos juegos de reglas aplicadas fueron obtenidas: a) El primer juego, contiene todas las reglas que han sido generadas desde la ejecución de 181 frases (se obtuvieron 248,800 reglas aplicadas). b) El segundo, solamente contiene las reglas que forman el árbol sintáctico. En este caso, todas las frases del corpus fueron analizadas (se obtuvieron 7.746 reglas). Debido a los dos juegos de reglas aplicadas, dos tipos de probabilidades de aplicación de las reglas fueron obtenidos.

Para el cálculo de la **probabilidad de que un rasgo aparezca instanciado**, se aplicó la fórmula descrita anteriormente, lo que dio lugar a un sistema regular de ecuaciones lineales, el que consta de 258 ecuaciones. Debido a los dos tipos de probabilidades de aplicación de las reglas obtenidos, dos sistemas de ecuaciones fueron generados. Ambos sistemas fueron resueltos por el programa matemático Octave⁴, dando lugar a dos soluciones, los que corresponden a las probabilidades de que un rasgo aparezca instanciado⁵. En el informe [LAS96] se muestra la resolución para cada uno de los sistemas de ecuaciones hallados.

A continuación, se presenta una gramática ejemplo (Fig. 4), el sistema de ecuaciones lineales (Fig. 5) obtenidas, y finalmente la resolución del sistema (Fig. 6).

⁴ El program matemático Octave fué desarrollado por John W. Eaton.

⁵ Más adelante, la solución que tiene las más altas probabilidades fueron consideradas para seleccionar rasgos.

- (r1) $A [x +] \rightarrow B [x @], C [y -]$ (0.1)
 (r2) $A [x @] \rightarrow C [x @, y +], B [x @]$ (0.2)
 (r3) $B [x @, y +] \rightarrow A [x @, y +], B [x +], C [x +, z @]$ (0.4)
 (r4) $B [x +] \rightarrow B [y @], C [x +]$ (0.3)

Figura 4: Gramática ejemplo, y las probabilidades de aplicación de sus reglas.

Las reglas de la gramática son: r1, r2, r3 y r4; las categorías: A, B y C; y los rasgos: x, y, z. Las probabilidades de que cada rasgo de cada categoría estén instanciados en la parte derecha de las reglas de la gramática son: $P(A, x)$, $P(A, y)$, $P(B, x)$, $P(B, y)$, $P(C, x)$, $P(C, y)$ y $P(C, z)$; las probabilidades de las reglas: $P(r1)$, $P(r2)$, $P(r3)$ y $P(r4)$. El sistema de ecuaciones lineales que se obtiene se muestra a continuación:

$$\begin{aligned} P(A, x) &= P(r3) P(B, x) & P(C, x) &= P(r2) P(A, x) + P(r3) + P(r4) \\ P(A, y) &= P(r3) & P(C, y) &= P(r1) + P(r2) \\ P(B, x) &= P(r2) P(A, x) + P(r2) P(C, x) + P(r3) & P(C, z) &= 0 \\ P(B, y) &= 0 \end{aligned}$$

Figura 5: Sistema de ecuaciones lineales a partir de la gramática ejemplo.

Para resolver el sistema de ecuaciones, se utilizó el programa de libre distribución Octave.

$$\begin{aligned} P(A, x) &= 0.18908 & P(B, x) &= 0.47269 & P(C, x) &= 0.53782 \\ P(A, y) &= 0.40000 & P(B, y) &= 0.00000 & P(C, y) &= 0.30000 \\ & & & & P(C, z) &= 0.00000 \end{aligned}$$

Figura 6: Solución para el sistema de ecuaciones de la gramática ejemplo.

4. Experimentos realizados para seleccionar restrictores adecuados

4.1 Planificación de los experimentos

Los resultados de la estimación de los parámetros del modelo realizado con la gramática de amplia cobertura permitieron seleccionar rasgos que tienen probabilidad alta de estar instanciados. En la tabla 1, se muestran éstos rasgos con sus probabilidades, por categorías. Con éstos rasgos se formarían restrictores para la realización de experimentos. La finalidad de los experimentos fué ver cuales de los restrictores propuestos son adecuados y qué características definen a éstos restrictores. Los experimentos consistían en el análisis de 30 frases seleccionadas desde el corpus del ALVEY, con cada uno de los restrictores propuestos.

FRASE	begap	conj	ellip	coord	fin	vform	agr
[we,help,out]	1384	1384	1858	1542	1968	2847	1352
[we,ask,out,the,abbot]	1867	1867	2499	2183	2456	3629	1993
[we.answer.to.him.for.mv. action]	2559	2559	3191	3191	3581	5349	2530

Tabla 3: Pruebas de restrictores con un rasgo para la categoría vp.

Las conclusiones que se obtienen de los resultados de éstos experimentos son: 1) Los rasgos de restrictores con los que se obtienen los mejores resultados (menor número de estados), están instanciados en la parte izquierda y derecha de todas las reglas de la gramática. Los restrictores con éstos rasgos son los adecuados. En la gramática que usamos éstos rasgos son: begap, conj y ellip de la categoría s; los rasgos begap y conj de vp; y conj y spec de n2.

2) Restrictores de rasgos no instanciados, en general no dan buenos resultados. Adicionalmente, se ha observado que en algunos de éstos rasgos, y que han tenido análisis eficiente, se debió a que en la operación de predicción del análisis en cada juego de estados, la estructura que contiene éstos rasgos subsume a predicciones posteriores, generando menos estados predictivos.

En restrictores de rasgos que tienen valores variables y que no generan análisis eficiente se ha observado que las estructuras de rasgos no subsumen a predicciones posteriores. Por ejemplo la estructura que contiene el rasgo *wh* en la categoría *s* generalmente subsume a predicciones posteriores, mientras que con *ever* en la misma categoría no; el restrictor que contiene *wh* siempre es mejor que el que contiene a *ever*. Sin embargo, es difícil saber cuando una estructura de rasgo va subsumir a predicciones posteriores, porque depende de la gramática y de la frase.

La tabla 4 muestra las pruebas de restrictores con dos rasgos para las categorías *s* y *vp* respectivamente, para dos frases. Las conclusiones que se obtienen de éstos experimentos son: al combinar rasgos de restrictores adecuados obtenidos en los experimentos de restrictores que contienen un rasgo, siguen estableciendo restrictores adecuados. Sin embargo al combinar cualquiera de éstos rasgos con otros de restrictores no adecuados, prevalece la no eficiencia en éstos experimentos. Por ejemplo en la tabla 3, se observa que, en la categoría *vp* los rasgos *ellip*, *fin* y *vform* forman restrictores no adecuados; en la tabla 4, se observa que al combinar cualquier restrictor que contiene éstos rasgos con otros restrictores, el proceso de los análisis son inefficientes.

Los resultados obtenidos de los experimentos con tres rasgos para todas las categorías similares a los resultados obtenidos de los experimentos con dos rasgos.

FRASE	categoría s Segundo rasgo						categoría vp Segundo rasgo					
	Primer -----						Primer -----					
	rasgo	conj	ellip	ub	ever	wh	rasgo	conj	ellip	coord	fin	vfo
[we, help, out]	begap	1381	1381	1411	1484	1439	begap	1384	1858	1542	1968	28
	conj		1381	1411	1484	1439	ellip			1858	2421	29
	ellip			1411	1484	1439	coord				2119	29
[we, ask, out, the, abbot]	begap	1863	1863	1909	1979	1892	begap	1867	2499	2183	2456	36
	conj		1863	1909	1979	1892	ellip			2499	3060	42
	ellip			1909	1979	1892	coord				2758	39

Tabla 4: Pruebas de restrictores con dos rasgos

En experimentos, con restrictores de rasgos que no tienen probabilidad alta de estar instanciados; pero se obtienen buenos resultados para ciertas frases; se observa que en frases que dan análisis eficiente, en la operación de predicción del análisis en cada juego de estados, las estructuras que contienen éstos rasgos subsumen a predicciones posteriores. Mientras que en frases que no generan análisis eficiente, se ha observado que las estructuras de rasgos no subsumen a predicciones posteriores. Algunos de éstos rasgos son *prd* y *aux* de la categoría vp, *begap* de la categoría n2.

Las conclusiones de los resultados de los experimentos realizados con restrictores de rasgos no instanciados, son similares al de los experimentos con restrictores de rasgos que proporcionan análisis eficientes para ciertas frases, es decir los resultados son buenos sólo para ciertas frases y la explicación es la misma. Algunos de éstos rasgos son *prd* y *past* en la categoría s, los rasgos *per* y *plus* en n2.

4.2.2 Segunda etapa de los experimentos con restrictores dinámicos

En la segunda etapa, el objetivo de los experimentos fué utilizar distintos restrictores en cada categoría. Los experimentos se empezaron con los mejores restrictores, en cada categoría, obtenidas en la primera etapa; los rasgos que contienen éstos restrictores están

instanciados en todas las reglas de la gramática. Posteriormente los experimentos se extendieron a restrictores de rasgos sin instanciar.

Las conclusiones que se obtienen de los resultados con éstos experimentos son: 1) Cualquier selección o combinación de los restrictores adecuados obtenidos en cada categoría dan siempre análisis eficientes. Parte de éstos resultados se pueden ver en la tabla 2, en las columnas de los rasgos begap o conj o ellip, ya que se obtienen los mismos resultados como en la primera etapa. 2) restrictores que contienen rasgos sin instanciar, no dan análisis eficientes, y al combinar éstos rasgos con otros se mantiene la no eficiencia.

4.2.3 Experimentos con restrictores generales

En experimentos con restrictores generales, el restrictor es el mismo para todas las categorías de la gramática. Para éstos experimentos, se seleccionaron los rasgos de restrictores adecuados por categorías; se ordenaron los rasgos por el número de veces en que aparece en cada categorías, esto es: 1º conj (3), 2º begap (2), 3º ellip (1), y 4º spec (1), el rasgo ellip no se consideró porque no da buenos resultados con la categoría vp.

En éstos experimentos se observan que los restrictores que contienen los rasgos conj y spec son los que dan mejor eficiencia en el análisis. Con restrictores que contienen el rasgo *begap* en la categoría n2 se obtienen buenos resultados sólo para ciertas frases. La tabla 5, muestra los resultados con uno y dos rasgos para las frases del ejemplo, además en ella hemos resaltado en donde los análisis no son los más eficientes. Dos características importantes de los rasgos conj y spec es que éstos rasgos corresponden a la gramática y no aparecen en el léxico, otra es que siempre están instanciados en todas las reglas de la gramática.

FRASE	conj	spec	begap	[conj. begap]	[conj. spec]	[begap. spec]
[we,help,out]	1377	1384	1381	1377	1377	1381
[we,ask,out,the,abbot]	1857	1867	1863	1857	1857	1863
[she,agrees,with,him]	1721	1730	1770	1765	1721	1766
[we,answer,to,him,for,my, action]	2545	2559	2597	2589	2545	2593

Tabla 5: Pruebas de *restrictores generales* con uno y dos rasgos

5. Conclusiones

Para nuestro estudio, acerca de cómo seleccionar restrictores adecuados cuando se utiliza la técnica de restricción, con gramáticas de unificación de amplia cobertura, hemos realizado diversos tipos de experimentos y con diferentes restrictores propuestos. La finalidad de ellos era observar cuáles de esos restrictores son los *adecuados*, y las características que presentan. Para ello, definimos un modelo estadístico, que nos permitiera conocer la situación de instanciación de los diferentes (270) rasgos de la gramática, para establecer restrictores candidatos a ser adecuados.

Dos clases de restrictores hemos usado, *restrictores generales*, en éste caso el restrictor es el mismo para todas las categorías; y *restrictores dinámicos*, en donde el restrictor puede ser distinto para cada categoría. Desde la valoración de los resultados obtenidos en todos los experimentos realizados, podemos destacar algunas características de los rasgos que han establecido *restrictores adecuados*:

- 1) En cada categoría de nivel superior de la gramática (s, vp, n2), hay un conjunto de rasgos, los que están instanciados en todas las reglas de la gramática, con los que se obtienen los mejores resultados, y con los cuales se establecen restrictores generales adecuados. En la gramática que usamos son: conj y spec.
- 2) Se ha observado que hay rasgos que son buenos candidatos en algunas categorías, pero en otras categorías no. Por ejemplo el rasgo begap es buen candidato en las categorías s y vp; pero en n2, no lo es. Por esta razón, extendiendo el concepto de restricción para que las categorías puedan tener diferentes restrictores (dinámicos), y asignando a cada categoría, restrictores cuyos rasgos son buenos candidatos, se han obtenido análisis muy eficientes de esta forma.
- 3) En restrictores dinámicos, se puede establecer una gama de restrictores adecuados. Por ejemplo, en la gramática que usamos, los rasgos que son buenos candidatos en sus categorías son: en la categoría s, begap, conj y ellip; en vp, los rasgos begap y conj; en n2, los rasgos conj y spec. Con los que se pueden formar, al menos doce ($3 * 2 * 2$) restrictores dinámicos adecuados, combinando cualquiera de esos rasgos de sus categorías.

Finalmente, se ha observado que los rasgos no instanciados no forman restrictores adecuados, en general. Además, al combinar esos rasgos con otros, siempre prevalece la no eficiencia en el proceso del análisis.

Referencias

- [CAR93] Carroll J.(1993); Practical Unification-Based Parsing of Natural Language. PhD thesis, Cambridge University,UK.
- [EAR70] EARLEY J.(1970); An Efficient Context-Free Parsing Algorithm. Communications of the ACM.
- [GER89] Gerdemann D.(1989); Using Restriction to optimize Unification Parsing. In Proceedings of the 1st International Workshop on Parsing Technologies, Pittsburgh, PA.
- [GER91] Gerdemann D.(1991); Parsing and Generation of Unification Grammars. PhD Thesis, University of Illinois, Technical Report CS-91-06 of The Beckman Institute.
- [GRO93] Grover C. & Carroll J. & Briscoe T.(1993); The Alvey Natural Language Tools Grammars(4th release). Technical Report N° 284, Computer Laboratory, Cambridge University,UK.
- [LAS95] La Serna N. & Díaz A. & Rodríguez H.(1995); Primeras aproximaciones para seleccionar restrictores óptimos en formalismos gramaticales de unificación para obtener analizadores eficientes. Informe de Investigación UPV/EHU/LSI/TR15-95, Universidad del País Vasco, Spain.
- [LAS96] La Serna N.(1996); Selección del restrictor adecuado en gramáticas de unificación de amplia cobertura. Informe de Investigación UPV/EHU/LSI/TR15-96, Universidad del País Vasco, Spain.
- [SEI87] Seiffert R.(1987); Chart Parsing of Unification Based Grammars with ID/LP Rules. Ewan Klein and Johan Van Benthem, editors, Categories, Polymorphism and Unification, Edinburgh/Amsterdam.
- [SHI85] Shieber S.(1985); Using Restriction to Extend Parsing Algorithms for Complex Feature Based Formalisms. In ACL Proceedings, 23rd Annual Meeting, University of Chicago. Chicago, ILL.
- [SHI86] Shieber S.(1986); An Introduction to Unification Based Approaches to Grammar. CSLI Lecture Notes N° 4, University of Chicago.Press, Chicago, ILL.